DST SEMINAR ON

Application of Computers to Bibliographical Information Processing. Some Developments in India (Bangalore)(10-13 July 1978)

BIBLIOGRAPHIC INFORMATION RETRIEVAL SYSTEM USING FORTRAN

ASIM KUMR PAL, Documentation Research & Training Centre
Indian Statistical Institute, Bangalore 560 001.

0 INTRODUCTION

Processing of bibliographic information is mostly a string manipulation and matching operation. For this purpose, data processing and string manipulating languages, such as COBOL, SNOBOL, PL/I, LISP, BASIC, etc are suitable. Using languages developed for scientific and numerical computation, such as FORTRAN, to bibliographic information processing has its own disadvantages. It is a problem-oriented, or more specifically, numerical problem-oriented language. Thus, it has its basic deficiency in dealing with string manipulation, comparison of two strings, etc. But this is not to say that FORTRAN should be avoided for this purpose- Because FORTRAN has its own advantages over other languages, insofar as it is considered to be a universally accepted language. Also, its portability is very high.

Asim Kumar Pal

In this paper, a case study of developing a bibliographic information retrieval system based on earlier models (1,2) in FORTRAN-IV is presented. Also presented are the details of additional facilities incorporated in the system,- together with major modifications or changes introduced in the 'search phase' of the system

I SALIENT FEATURES OF THE SYSTEM

II Input Processing Stage

At this stage, the present system essentially follows in principle the old system (l). But it extends the scope of updating and data validation to a large extent. The level of updating in the present; system does not restrict to the card level, but extends to the level of the record which consists of a number of cards, eg dropping of a bibliographic record, replacement of a bibliographic record, etc. Moreover, an error key has been introduced in the card images on the master file. A comprehensive sex of error codes has been designed to indicate errors in the input card sequence, update card, input card for new bibliographic records - all at the same time.

12 Data Base Creation Stage

At this stage too, the present system does not differ much from the old systems (1,2). The 'facet structure' field has been dropped in the present system considering its unsuitability in most of the schemes for classification. Due to the inability of FORTRAN

to handle binary numbers, 'sequential search' has been used on an 'ordered set' of characters in order to construct the 'ordinal value' field. Therefore, each character in the 'class number' of the input record is replaced by a decimal number. Moreover, 'unnecessary' blanks have been suppressed before the construction of the 'directory of tags' and the 'ordinal value' field.

13 Searching and Outputting Stage

This is the stage where almost all the changes are fundamental — starting from the searching and retrieving strategy to the outputting strategy. Two different search programs — CLSRCH and BOOLSRCH — have been developed, one for the 'class number' approach and the other for search on Boolean combination of different elements from different fields. The CLSRCH program is almost identical to its counterpart in the old system (1). But the BOOLSKCH program is different in almost all aspects from its counterpart in the previous system (2). Here, direct access methods have been used with. maximum efficiency. All the queries (a maximum of 1000 queries, each with a maximum of 44 retrieved records and the total number of retrieved records being less than 5667 for the IBM 360-44 system with two-disk data sets, each of 5000 records of 360 bytes) can be searched in *one* pass of the data base file on tape/disk. The search expression is to be input in a highly coded form, unlike the previous system (2). The printing of the output may be formated by the user or the query maker, whereas in the old system, the print format was recorded

in the input of the bibliographic record itself.

2 INPUT PROCESSIMG

21 Input Format

The bibliographic data is punched onto cards from column numbers 1 to 74 continuously with the fields delimited by field delimiter symbol and end of the record marked by record-delimiter symbol. Column numbers 75 to 76 contain the record (document) number; column number 79 contains the serial number within a record (card sequence number). Column number 80 contains the 'update key (UK) for the update run; otherwise it is kept blank.

22 Card-to-Tape Updating & Data Validation

The bibliographic data punched in the punch cards (80 columns) with record number and card sequence number are read by a program CTUDV. The cards are checked against certain types of errors and appropriate error messages against error message codes are printed. A file -- Old Master File (OMF) — of bibliographic data cards -- is created on tape. The same program is used to correct the errors in the card images in the OMF (through addition to-, delition from, or changing the existing card images)_by giving correct cards as input to the update run of the program. The update keys used are described below.

221 Table of Update Keys

Update Key	Description
1	Replacement of a bibliographic record
2	Dropping of a bibliographic record
3	Dropping of a bibliographic card
4	Setting the 'error indication key' (IERMKT) in the card image in CMF to O.
5	Replacement of a bibliographic record
6	Insertion of a bibliographic record/card
0,	Addition of a new bibliographic record

The output of the update run Is a new. master file (NMF) on tape. The format of the card images in the GMF/ NIT is the same as that of the input data cards, except for the 80th column which contains '0' or '1' (error indication code - IERMKT), depending upon whether the card has any error (possibly) or not. All the card images in the NMF are in the increasing sequence of the record number and the card sequence number within a record

The following are the specification checks made on the input data.

1) Number of fields present in a bibliographic record;

2) Presence of 'record delimiter symbol' in a bibliographic record;

3) Sequence, of record (document) number corresponding to each bibliographic record;

4) Card sequence within a record; and

5) Consistency of the update key and the sequence of the update cards as related to the sequence of the card images in the OMF.

The card images of the erroneous cards are printed, with the position of errors indicated by asterisks in the appropriate positions along with the corresponding error codes. The following table gives the error codes and the corresponding error messages, together with the , action taken by the program CTUDV,

222 Table of Error Message Codes (CTUDV)

SN	Error Code	Source of Error	Information printed,if any	Action taken
			4	
1	99	Error in the card sequence	Card image (input)	Frocess next input card
2	1	Error in the update key (range exceeded)	-do-	-do-
3	2	Inconsistency of the record number in card with that in card image in OKF (UK is 3, 4 or 5)	-do-	•-do-
4	199	Inconsistency of the cerd serial number with that in card image in OMF (UK Is 3, 4 or 5)	-do-	Set IERKRK=1 process the next input card

4

5	200	IERMKT=0, but UK is 3, 4 or 5	Tape image (OMF) and card image (input)	Set IERMKT=1, continue with present update card
6	202	Record deleted (message), (UK=2)	Record number of the deleted record	Delete the record from OMF and process the next card (input)
7	3	Record for deletion not in OMF (UK=2)	Card image	Process next input card
6	4	Serial number of input card and that of tape image is the same (UK=6)	-do-	-do-
9	5	Input card record number is less than OMF record number (for the first time while inserting a card, UK=6)	-do-	-do-
10	6	Record with the same record number is in OMF (it is not an update run; UK=0;	-do-	-do-
11	7	Record with the same record number not in OMF (UK=1)	-do-	Inssrt all cards of this (bibliographic) record in the NMF. Check them and give necessary messages.

1			4	
12	8	Input card deck is exhausted (message)	Tape mmage and card image	Copy the rest of records from OHF to NMF
13	800	Records in CMF are exhausted (message)	-do-	Process all the rest input cards. Check them, give error messages and put them **in** NMF .
14	9	Copying of a record from OMF onto NMF, even if IERMKT=1	Tape image	Set IERMRK=0, copy this record from *Cl'J* to NMF Continue with xhe next card-reccrd in CMF
15	10	UK is greater than one for new cards which are not update cards	Card image	Process the next input card
16	11	Input ,'document type' code is wrong, or number of record delimiters or field delimiters exceeds their limits	-do-	Put the error symbol in the current position. Set IERMKT=1 and continue.
17	1101	Card serial number within a record in error	Expected sl.nc. and actual sl.no of the card and the card image	Continue processing

18	1100	Number of fields / 11 (or, any other pre-specified number), and/or the number of record delimiter symbol(s) £ 1	Number of fields and record delimiter symbol(s)	Continue processing
19	12	Error in card seouence	card image	Process the next input card

Note.- 1. Card image - image of the input card, *
Tape image - image of the card-record in OMP or simply, CMF card,
Input card - the card that just has been read from the deck of input data cards,
OMF card - the card-record image that jgst has been read from the OMF.

2. I/O operations that are done on tape can equivalenxly be done on disk, using it as a sequential access device and this can be done vrithout any change in the source program, but only in the job Concrol Lanauge (JCL) cards.

3 DATA BASE CREATION

The corrected card images on tape are read by the program GEMDAB and a data base for bibliographic records is created on tape The format of the records in the

data base is similar to that of CORC records *(2)*.• But due to the limitation of FORTRAK-IV, the 'ordinal value' field is constructed using ? 'sequentied match' or 'sequential search' method on the 'set of ordered characters' (which is the set of characters used in the class number fields when they are arranged in the sequence demanded by the scheme for classification concerned). instead of using 'Table Look Up' technique as used in the eld system (1,2). The 'facet structure' field earlier used in the programs CCRC (2) and BlRS-1900 (1) has been dropped in the present system, as it is unsuitable in most of the schemes for classification other than the COLON system. An additional processing of 'blank suppression' has been introduced. 'Unnecessary blanks' are suppressed so that oetween any two strings of characters, there is only one blank character and no punctuation character is followed or preceded by blank characters. This is done before the construction of 'ordinal value' field and the creation of 'directory of tags' (which are nothing but the pointers to the starting locations of fields). Also, the existence of any 'illegal character' (character outside the 'set of ordered characters', in the 'class number' field flags an error ana appropriate error messages are given. 'Ordinal value' is constructed based only on a certain number of characters starting from the beginning of 'class number' field. If the number of fields in a bibliographic record is more than a specified number (to be input through a control card) tags are created only for the specified number of fieldc. And if it is less, tags for the

remaining fields remain zero. However, in either case, error messages are provided. Each of the logical records in the data base contains the record (document) number, 'directory of tags' and bibliographic data as a continuous string of characters.

This program GENDAB requires the 'set of ordered characters' to be input through a control card.

Note.- In this program also, the I/C units can be tape/disk as is available.. In any case, access is sequential. No change is required in the source program.

4 SEARCHING THE DATA BASE .

The data base created in the earlier step (that is, by the program GENDAB) is searched by two different approaches by two different programs'. One approach is purely a 'class number' search approach and the other one is a search using the 'Boolean combination' of 'class number', 'name of the author', 'key words', etc.

41 Class Number Search

The class number search program (CLSRCH) searches the data base using the class number provided in the queries. It creates a tape file containing the retrieved records with corresponding 'query identification' ana 'print format codes' supplied in the query. If the 'class number' In a query terminates with a 'field delimiter' symbol, it is matched with each of the class numbers in a bibliographic record until a match is found

or end of the 'class number' field is encountered and then the record is retrieved if there is a complete matching. If the 'class number' in a query terminates with a 'truncation symbol', the match is required only upto the truncated position. Determined by the availability of the memory size, a number of queries are processed in a single pass of the data base file,

The format for the query card input is given below,

411 Query Card Format (CLSRCH) Table

Card Col No.	Field Description
1-8	Name of the organisation of the user
9-24	Fame of the user
25-26	Sequence number of the query for this user
27-38	Codes for 'print format'
39-78	Class number

Note.- The 'print format codes' incidicate which are the fields to be printed in seperate lines (or paragraphs;, Assuming the number of fields to be twelve, twelve columns on query card from Col No. 27 onward 'have been allocated for print format — one- column for each field. A '1' indicates that the corresponding field should be printed on a seperate line and a '0' indicates otherwise.

All the retrieved records along with the corresponding 'query details' arc written on output file.

A logical record of this file contains the 'query identification' information (it consists of the 'name of organisation' field, 'name of the user' field, 'sequence number' of the query), 'print format codes* end the retrieved bibliographic record. This program also prints the number of retrieved answer for a particular query along with the query card image, It also checks the validity (non-blank characteristic) of the 'user identification' field. And any error in this validity check rejects the query input card. Moreover, the printing subprogram (SPRINT) can print the output directly on the printer following , the 'print format codes'. But, in that case the answers printed will not be in the sequence of the 'user identiiication'. Instead, the output file on tape can be sorted using a sort routine according to the 'user identification' key and then the sorted answers can be printed by the printing subprogram SPRINT.

42 Boolean Search

421 Control Information Input

The Boolean search program (BOOLSRCH) can be used to search the data base on any field and also a Boolean expression (or combination) of words (query terns) occurring in different fields. The program requires two control cards to be input. The first one contains truncation symbol (use-d to truncate the query or search elements), field delimiter symbol and a set of punctuation characters (used to seperate cut the data elements for matching). The layout of the second control card is given below.

4211 Control Card (2) Format (BOOLSRCH) Table

Card Column Number	Field Description
1-2	Maximum length of a 'search element[1]
3-4	Number of 'punctuation characters'
5-6	Maximum number of distinct 'search elements' in a 'search expression'
7-8	Maximum 'length' of a 'search expression[1]
9-28	Lower/Upper limits of each of the coordinates of the 'five points code'
29-32	Lower/Upper limits of 'field number' in a bibliographic record
33-36	Maximum length of a bibliographic record (excluding the "directory of tags' and 'ordinaI value') that can be put in a physical record on disk
37-40	Maximum length of a bibliographic record
41_44	Laximum size of a physical record on disk
45-43	Maximum number of records on a data set on disk

Note- - 'The control information for the last four field in the above table is specific to the IBM 360/44 data set- and disk organisation. This information is completely machine dependent. Also, corresponding changes in the I/C operations may have to be incorporated to suit BOOLSRCH for a different machine configuration.

422 Query Card Input

The Input for a query consists of three sets of cards. The first set has one card only, which contains

'user identification', 'print format codes' and 'document type' codes. The second set also has one card which contains the number of distinct search elements used in the query, the 'length' of the search expression and the search expression itself in a coded form. The third set of cards may consist of any number of cards, depending en the number of search elements. Each card in this set has six fields — two fields ('field number[1] in the bibliographic record occupying 2 columns and string for the search elements occupying *24* card-columns} for each of three search elements that can be punched in a card.

423 Query Formulation Scheme

The query is formulated on the basis of a narrative statement of the interest of the user. The scheme for query formulation, its logic and the coding of the formulated query has been described in the following text. The explanation has been substantiated by an example.

Let a user narrate his interest as follows:

!:I am interested to receive information about books and articles in periodicals on the role of energy on the social development and production. I am not interested in the articles published in the Journal of Energy Policy, nor in any article written by A.V. LAZER"

The query elements and their corresponding field numbers in which they occur, are first put in a table as illustrated be-low;

SN Field N . Query/Search £lement

Asim Kumar Pal

4231 Query Elements Table

SK	Field	N	Query/Search Element
1		1	620.9*
2		2	ENERGY#
3		2	SOCIAL DEVELOPMENT^
4		2	PRODUCTION
5		3	LAZER (A V)#
6		5	JL. OF ENERGY POLICY#

In this example, the type of documents (records) required is A or C. We have used in the above table, '*' and '# ' as the truncation and delimiter symbols,

Search expression for this query (in terms of SNs):
U OR. (2 .AND. (3.OR. 4))) .AND, .NOT. (5 OR, 6)
This expression has been resolved in the following way.

4232 Table of Resolution

Reference Code	Query Elements Combination (QEC)
1	3 iOR. 4
2	2 .AND. 1
3	1 .OR. 2
4	5 .OR. 6
5	

1

Note.- The underscore indicates the reference to a query elements combination instead of a. single search element..

424 Query Logic

The logic of the breaking down of a query expression has been based on the following (algebraic) recursive relationships that exist among the query elements combination (QEC), individual query element (QE) and the Boolean operation (B0) in a search expression (SE).

< 3E) = < QEC>
<QEC> = <QEC> <B0> <QEC>
<QEC> = < NOT> QEC
<QEC> = {x : x is any QE }
<NOT> = {.NOT.}
< B O > ={AND. . , .OR. }

In general, to evaluate any search expression it is needed to evaluate a number of QECs in succession. bach QEC, in its turn, is evaluated in terms of its two operands joined by a Boolean operation. Each operand itself is a pre-evaluated QEC, a pre-evaluated QE or a new QE, and each is possibly accompanied with a logical .NOT. Thus, any operand is one of the four types: (1) QE (2) .NOT. QE (3) QEC (4) .NOT. QEC. Depending on the type of operand, a code called 'Type code' is attached with each of the operands. In case it is of type (1) or (2), it is needed to give the QE

explicitly, or the position of the concerned QE in the list of query elements. The latter one is preferred for the sake of compactness of coding. Thus, there arises the need for 3 list of distinct query element to be input through data cards. The reference to the old/new QE or QEC is termed as 'reference code[1]. The 'operation code' of a QEC is 0 or 1, depending on the operation .OR. or AND. respectively, th?t joins the two operands.

Format of a five-point code ;
(Type code, Ref. code, Operation code, Type code,Ref .Code;

First operand Second operant!

Table of Five Points Code

QEC NO.	Five Points Code
1	1,3,0,1,4
2	1,2,1,3,1
3	1,1,0,3,2
4	1,5,0,1,6
5	3,3.1,4,4

It is evident from construction of.the QEC's that the logical value of the last GEO in the sequence will determine the relevance of a bibliographic record for the search expression concerned.

The limits of each 'coordinate point' of the five points code is determined as shown below;

4241 Table of Limits for the 'Five Points Code'

Bibliographical Information Processing

The limits of each 'coordinate point' of the five points code is determined as shown below;

4241 Table of Limits for the 'Five Points Code[1]

Coordinate Point No.	Lower Limit	Upper limit upper Limit
1,4	1	4 •
2;5	1	Maximum (Maximum length of an expression, maximum number of distinct query element,
3	0	1

The input to the 'formulated query' consists of cards having the details given in the 'Table of Query Resolution' and the 'Table of Limits for the Five Points Code'.

425 Query Processing

The query cards are analysed and appropriate error messages, if any, are printed. The error message codes displayed here are not absolute but flexible depending on the maximum length (l) of an expression ana and the maximum number (n) of distinct search elements allowed. The error codes will range-between 1 and (4+n51). This'has been done to pinpoint the error location.

If the maximum length of an expression is$_s$ say, 10, and the maximum number of search elements is, say, 9, then the set of error codes is as follows

4251 Error Table (BOOLSRCH)

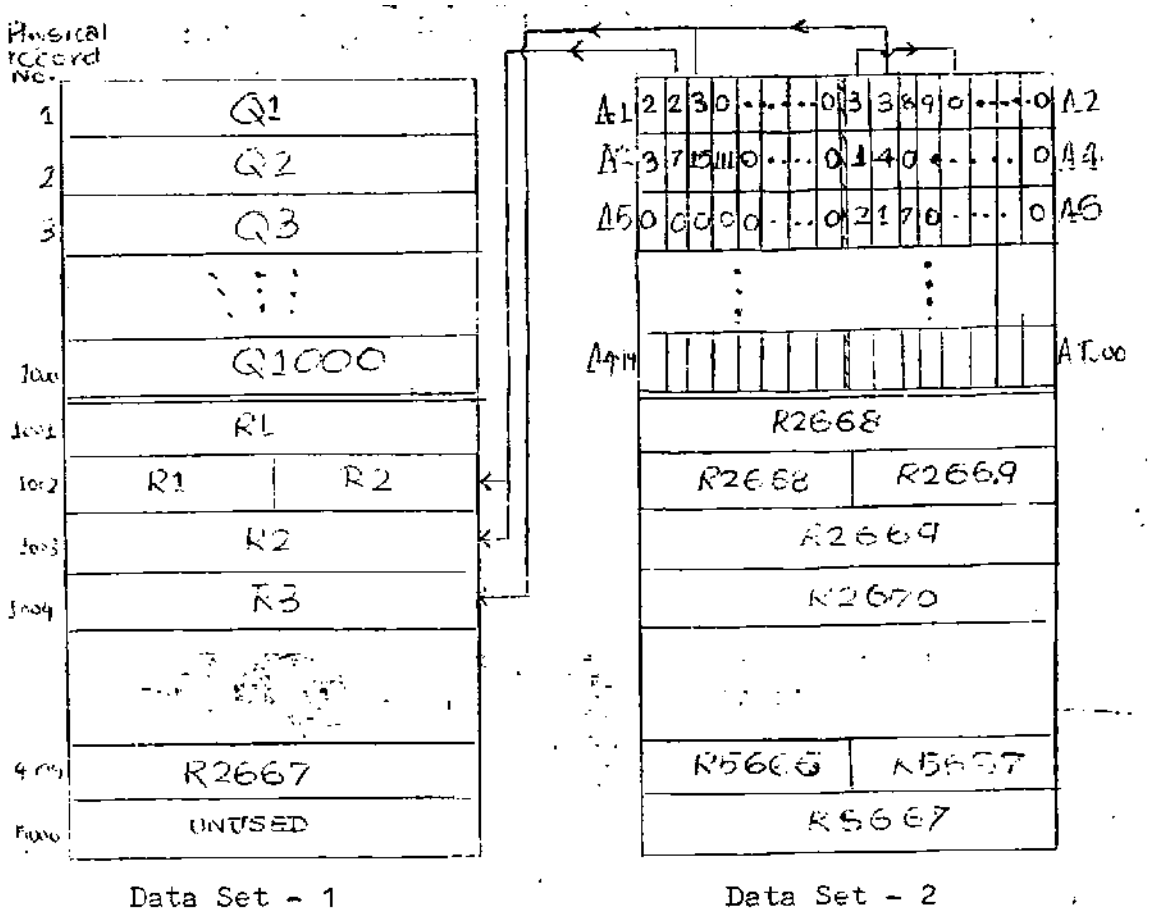
Error Code	Source of Error
1	First four characters in the 'name of organisation' field are blank.
2	First four characters in the 'name of"query maker/ user' field are blank.
3	Number of search elements is not within its limits.
4	Length of search expression is not within its limits.
5	Field number for search element 1 is not within its limits.
6	Field number for search element 2 is not within its limits.
13	Field number for search element 9 is not within its limits
14	Coordinate point 1 of QEC 1 is not withir. its limi-ts-
15	Coordinate point 2 of QEC 1 is not within its limits.
18	Coordinate point 5 of QEC 1 is not within its limits.
19	Coordinate point 1 of QEC 2 is not within its limits.
63	Coordinate point 5 of QEC 10 is not within its limits

Note.- The program BOOLSRCH finds out all possible errors within its scope. That is, it does not stop with the first error located. However, even a single error

rejects the query and it gives a print out of the rejected query image. But, if there is any error in giving the number of search elements, it may read -less or more number of cards than it is required for reading the search elements. Eventually, the input for an uncertain number of -queries following this may go without proper processing.

426 Search Procedure

The search procedure consists of three distinct stages, in which two data sets are used. Data set-1 has space allocated for storing the queries and retrieved records. Data set-2 has space allocated for storing directory of index to the retrieved records and for storing the retrieved records too. Thus, the retrieved records are stored in Data set-1 as -long as space is available and otherwise, they are put on data set-2. Figure 1 gives the organization of the two data sets.



Q1,Q2 ... Queries Stored in Data Set - 1

R1,R2 ... Retrieved Records on Data Set - 1 and Data Set - 2 with index number 1,2,...

A1,A2 ... Directory containing pointers to answers for Q1,Q2 ...

Fig 1: Organisation of Data Sets in BCOLSRCH

4261 Building the Query File

Each set of query cards is read and analysed for specification violations; The valid ones, i.e. error-free queries, are written on disk in 'data set-1'. At the end of this step, a file of valid queries will be available in the data set-1.

4262 Building the Directory for the Answers

A bibliographic record is read from the master file on data base file on tape/disk and it is matched with each of the queries in the query file built at the earlier stage.. If it matches with a particular query, the index number (index number of the ith retrieved record is i) of this particular record is written in . the next immediate field available in the record allocated for the query, in the directory for answers in data set-2. (See figure 1). In the present program, a physical record on disk has been allotted for' the directory of two queries.

4263 Constructing the Retrieved Records file

As and when a record is retrieved for a query, it is written in one of the two data sets depending on the availability of space. But in the present case, the maximum length of a physical record is 360 bytes due to the limitation of IBM 360-44. Hence two consecutively retrieved bibliographic records consisting of about 500 bytes, are written on three.physical records on disk.

Note.- 1. If there are n queries and m retrieved reccrds in the present context, approximately [3(m+n)/2] physical records would be required to allocate space on the disk.

2. The 'index number' for a retrieved record indicates the relative position of the record in the file of retrieved records. Thus 'index number' relates only to those records which are retrieved for the queries. A record which is not retrieved by any query is not included in this file.

427 Retrieved Records Output Procedure

The first record from the directory of index to the retrieved records pertaining to the first query is read. The index numbers contained in the list are the relative sequence numbers of the records (in the file of retrieved records) that are retrieved for the first query, oince the records are of fixed length, the index number directly determines the exact location of the record in the file. Using the index numbers, the records in the file are acessed successively and the records are printed and/or written on tape along with the 'query identification' details as required. This process is repeated for all the records in the directory and the retrieved records are printed and/or written on tape.

The retrieved answers to a particular query will be together in the print out/output tape and for the different queries, they will be in the same sequence as that of the input queries. This procedure elimi-

nates the need for sorting of the retrieved records, to bring the answers to a query together, or to maintain the sequence of the answers as that of the queries.

428 Features of the Search Process

1. A query element or search element may be purely a single word (a word is a string with no 'blank' in between) or a compound word consisting of more than one words joined by different 'punctuation' characters or blanks, e.g: (a) Engineering, (b) Engineering Industry, (c) Why Mathematics? However, it must terminate with an end/truncation symbol.

2. Search is possible for a complete string or a truncated string.

3. The query maker/user may give his option of making paragraphs for the fields, according to his choice, in the printout of the retrieved records to his query.

4. Searching of the query terms in a search expression is made only when it is a logical necessity. For example, if there is an AND, between two search elements and the first one is absent in a record, the combination is automatically assumed to be absent in the records.

5„ Logical duplication in evaluating the status of a search expression as related to a bibliographic record is avoided. For example, if the same QE or GEC is referred to in the search expression more than once, actual search through a bibliographic record will be made only once.

6. Search can be made on one search element too. To search the QE 'A', one may search by the expression 'A.AND.A' or 'A OR. A' .

5 SCOPE AND LIMITATION OF THE SYSTEM

Even though the programs developed for this system as mentioned earlier have been written in FORTRAN-IV, an attempt has been made to write the programs in as much a structured form as possible, so that understanding of the basic philosophy behind the programs and the system is easy and clear and one can translate the programs into other languages without much difficulty. Or to increase the efficiency, one can write the programs in more than one language, and then have the load modules (in machine language) of the different programs link-edited and execute this one at different stages with different control cards. One may change the programs appropriately to process different types of inputs, or under the different restrictions and limitations imposed by a different machine. Due to lack of time and opportunities, many modifications and additions which were perceived to be Introduced even in the present programs were not possible. Some of the possible modifications and extensions are outlined below:

l.'Even though the programs are written in the FORTRAN-IVj It is possible to retranslate the programs into AUTOMATE (Honeywe11-400) and FORTRAN-II with little difficulty. The logical variables in the program, 3Q0LSRCH can be replaced by the variables of integer

type taking the values '0' and '1'. If any installation does not have the direct access devices, the BOOLSRCH program can use the algorithm to that used by the CLSRCH program. And then, of course, it is required to sort the output file on tape according to the 'query identification' key.

2. To make the input at the query search stage easier and to avoid the coded form adopted, for the query formulation and coding scheme in the BOOLSRCH program and to use the Folish notation, one can use any subprogram to convert an expression into its Polish form and then evaluate the expression. But it is to be noted that in FORTRAN-IV, this will introduce a degree of inefficiency in the system from the point of view of increased time required for searching, because the translation of the expressions into Polish notation and evaluating the Polish notation is added to the searching strategy. And also since FORTRAN-IV is a nonrecursive language with static storage allocation, it is a burden on the programmer and the requirement of the memory storage for the program. Keeping this in mind, to decrease the run time for BOOLSRCH program, it has been decided to input the expression in a highly coded form.

3- A number of statistical tables can be provided in the output of both the search programs. For example, [1) how many bibliographic records are retrieved for each query? ('Query answer frequency' table is formed in this process!J.

(2) to construct a table indicating how many timet- each bibliographic record has been retrieved. Arrange the bibliographic records in ascending sequence of their response capacity. ('Document response table' is formed in the process).

The 'query answer frequency' table will help the query maker to make his search expression more useful and accurate for a better response to his query. This can be used as a 'short display method' by showing the initial list on the C.R.T. screen. This is one step forward to the direction of an on-line S.D.I service. Because, looking at the initial list, the user, if he desires, may intend or extend the query and in that case, it may not be necessary to search the whole data base once again but only the already retrieved records needed to be searched.

The construction of 'document response table' will help to update the data base by deleting the infrequently retrieved records from the data base considering them either as obsolete or keeping the deleted records in a sepe_ate file ('file of rarely retrieved documents') to be-accessed only on special requirement. Similarly, it is possible to seperate out a 'file of frequently re- trieved documents' from the data base. And, for any given query search through this file should be made first and information' can be provided by 'query answer frequency' table, so that the user may intend or extend his query more appropriately before going to search the whole of the data base.

These.tables also give an indication to change the - portions of the data base from time to time and also to revise query formulation schemes and vocabulary control methods adopted by the information scientist, foote these two special files themselves have to be updated from time to time.

(k) Statistical tables may also be maintained to control the query formulation method and check the accuracy of the query formulation by the information scientist through a 'feed back system'. And, for this purpose, each answer is printed in two copies, one to send to the user and another to the information scientist concerned to keep him well informed of the status of the queries which he formulates. This part of the job can essentially be computerised, when the wnole SDI system will be on-line.

(5) Retranslation of the programs into C0B0L/PL1 is another immediate possibility. Some of the important things that can be achieved are:

a) Explicit use of date structures$_s$ e.g.

1) Data division in COBOL allows different levels of record structures.

2) PL/1 language has the capability of defining date structures including linearly linked structures.

•b" Character and bit manipulation of data. e.g.

1) PICTURE clause in COBOL can define any variable to be of any type and length. And also some COBOL compilers, as in ICL-1900 series, allows the definition of binary fields.

2) The DECLARE • clause allows the definition of any variable to be numeric-fixed or floating,binary, etc.

c) The file handling systems in COBOL are much more efficient as compared to FORTRANI-IV. e.g.

1) Programmer does not have to bother about the maximum size of a physical record on tape/disk.

2) Also, some easy types of updates are possible by the system itself. One does not have to write always an explicit program for updating, but'instead, simply insert a few instructions at appropriate places.

d) The direct access on disk is as possible in COBOL as in FOftTRAN-IV. To access a record, one may just give the record number to be accessed. But programmer may increase the efficiency of the retrieving strategy by calculating 'relative track address' for the record to be accessed, hote that, here calculation is based only on a sequential number, indicating their relative positions in the file.

e) For searching through a bibliographic record, the SEARCH instruction in COBOL can efficiently be used.

6; A program to update the data base at the level of individual fields can also bo developed. This will make updating the data base much more efficient and easy because one need not always gc back to 'updatin^ at the card level' stage and creating the data base once agair.

7) The execution of the CLSRCH program can be made faster if the 'direct access[1] method is inserted in the program as in the BOOLSRCH program and also the output

will automatically be sorted according to the 'user identification' field.

Similarly, BGCLSRCH can be made useful for 'physical sequential access devices' and sequential data organisation and sequential retrieval methods where direct access devices are not available. The data organisation and retrieval, technique in BOOLSRCH has to be changed to that in CLSRCH keeping the searching strategy (matching the bibliographic records with the queries; unchanged.

6 ACKNOWLEDGEMENT

My greatest acknowledgement is due to Shri FJ Devadason, without whose close guidance and constant encouragement this work would not have been completed and this paper would not have seen the light of the day.
I am very grateful to Prof A Neelameghan for his timely suggestions and guidance at different stages of the work.

7 BIBLIOGRAPHICAL REFERLNCLS

1. DLVADASON (F J). BIRS/1900 - Bibliographic Information Retrieval System: A set of programs for SDI and retrospective search services for an information service network. -(Paper submitted to the DST Seminar on Application of Computers to Bibliographical Information Processing: Some Developments in India (Bangalore; (1978;).

2. RAVICKAHDFU RAO (I K) and HEMALATHA IYKft. Computer-based SDI; COBOL program. (Lib Sc. 13:1976; Faper K).